Neural Academy

CORSO NEURAL ACADEMY 12-22 / 07-23

Alessandro Gazzillo

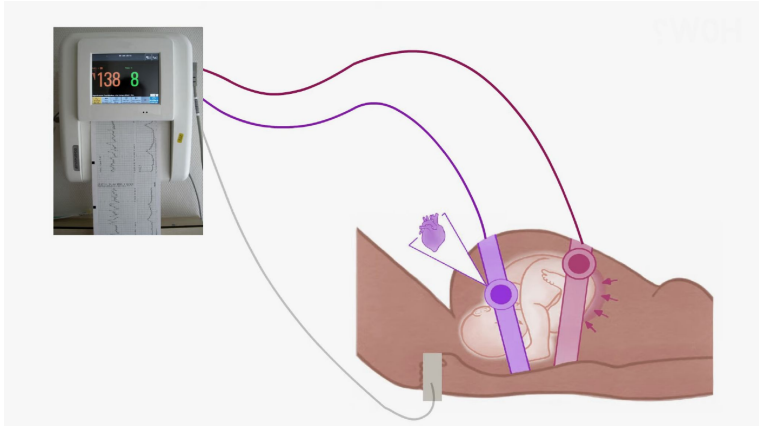
Fetal Healt Classification

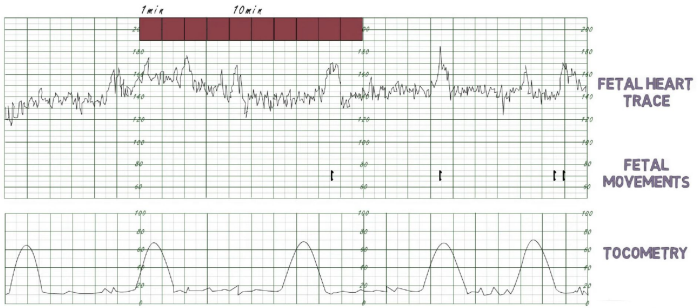
Classificazione della **Salute Fetale**

Lo scopo del compito è quello di **predire** e **classificare** la salute dei feti dalla 38a settimana di gestazione attraverso i **dati ricavati** dal un esame **cardiotocografico** della durata di mezz'ora; in tutto i feti possono essere classificati come "Normali", "Sospetti" e "Patologici".

Perché tutto questo, la **riduzione della mortalità infantile** è un indicatore **chiave del progresso umano**, la stragrande maggioranza dei decessi di bambini sotto i 5 anni di età si sono verificati in contesti con poche risorse e la maggior parte di questi potevano essere evitati, questo esame è poco costoso ed è accessibile a tutti, esso può predire con una certa accuratezza la salute del feto.

L'esame cardiotocografico consiste nel posizionamento di due placche (che possono emettere e registrare ultrasuoni) sulla pancia e una pulsantiera posta nella mano della mamma in gestazione, un elettrodo misura principalmente il battito cardiaco del feto, l'altro elettrodo misura le contrazioni dell'utero e la pulsantiera viene premuta dalla mamma quando percepisce dei movimenti del feto. Il Cardiotocografo stampa 2 istogrammi che verranno poi analizzati per ricavare gli stessi dati utilizzati da questo Dataset.

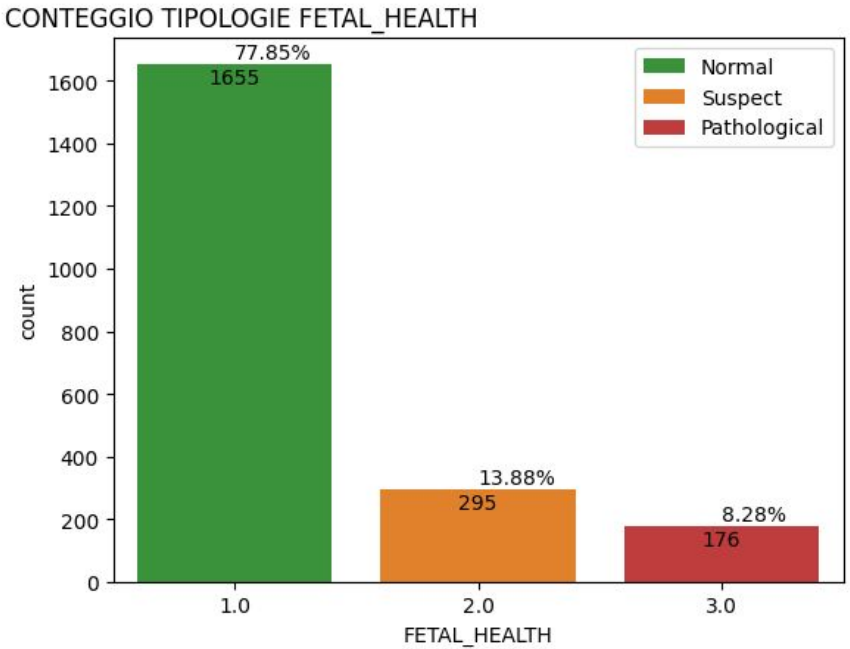




EDA - Exploratory Data Analysis

Il Dataset contiene **2126** osservazioni spalmate su **21 features**, queste features sono tutte numeriche e sono ricavate da un **esame statistico** dell'istogramma ottenuto dalla visita cardiotocografica.

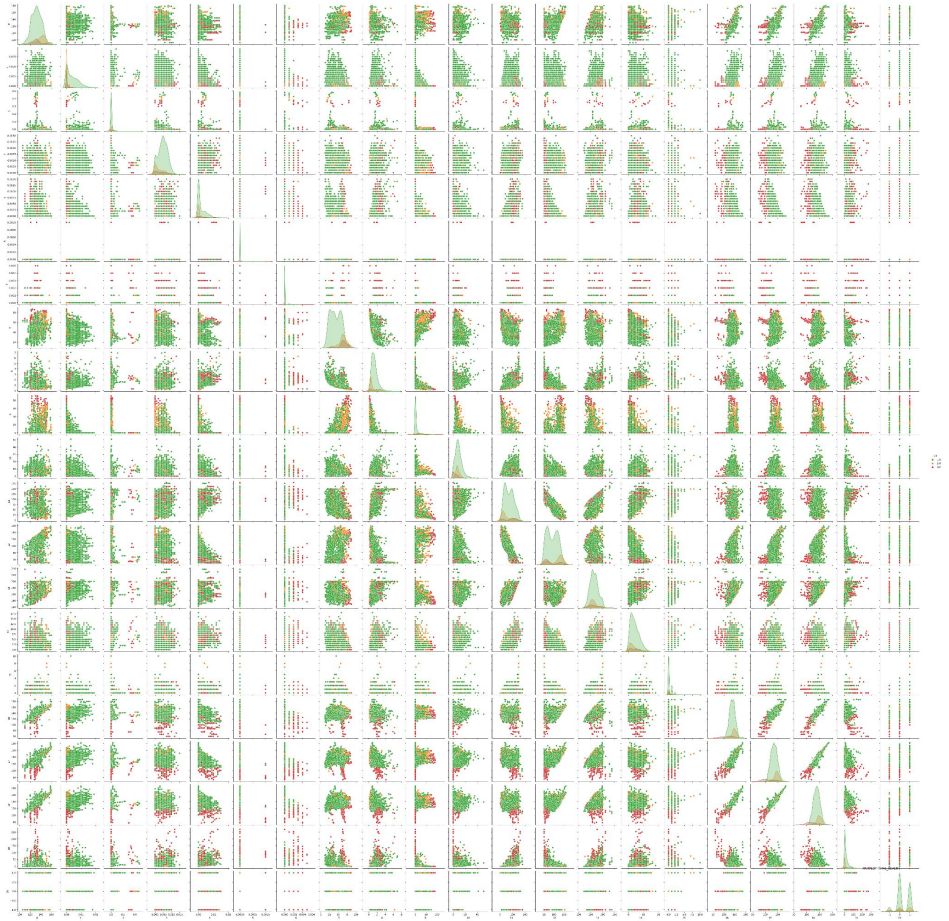
Rispetto al target il **dataset** è molto **sbilanciato**, il grafico a barre ottenuto dal notebook lo dimostra.



EDA

Nel **pairplot** è possibile notare dei cluster rossi (Pathological) e qualche accenno dei gialli (Suspect), si possono notare anche delle correlazioni lineari tra features direttamente connesse da ragionamenti statistici come la media, moda e mediana e anche la BASELINE_VALUE (che per definizione è una media del battito cardiaco). Può darsi che il modello KNN funzioni bene con il dataset.

Si possono notare anche delle features con quantità discrete di indice: **3, 4, 5, 6 e 15** riferite alle colonne.



LISTA DELLE FEATURES:

0 BASELINE_VALUE

1 ACCELERATIONS

2 FETAL_MOVEMENT

3 UTERINE_CONTRACTIONS

4 LIGHT_DECELERATIONS

5 SEVERE_DECELERATIONS

6 PROLONGUED_DECELERATIONS

7 ABNORMAL_SHORT_TERM_VARIABILITY

8 MEAN_VALUE_OF_SHORT_TERM_VARIABILITY

9 PERCENTAGE_OF_TIME_WITH_ABNORMAL_LONG_TERM_VARIABILITY

10 MEAN_VALUE_OF_LONG_TERM_VARIABILITY

11 HISTOGRAM_WIDTH

12 HISTOGRAM_MIN

13 HISTOGRAM_MAX

14 HISTOGRAM_NUMBER_OF_PEAKS

15 HISTOGRAM_NUMBER_OF_ZEROES

16 HISTOGRAM_MODE

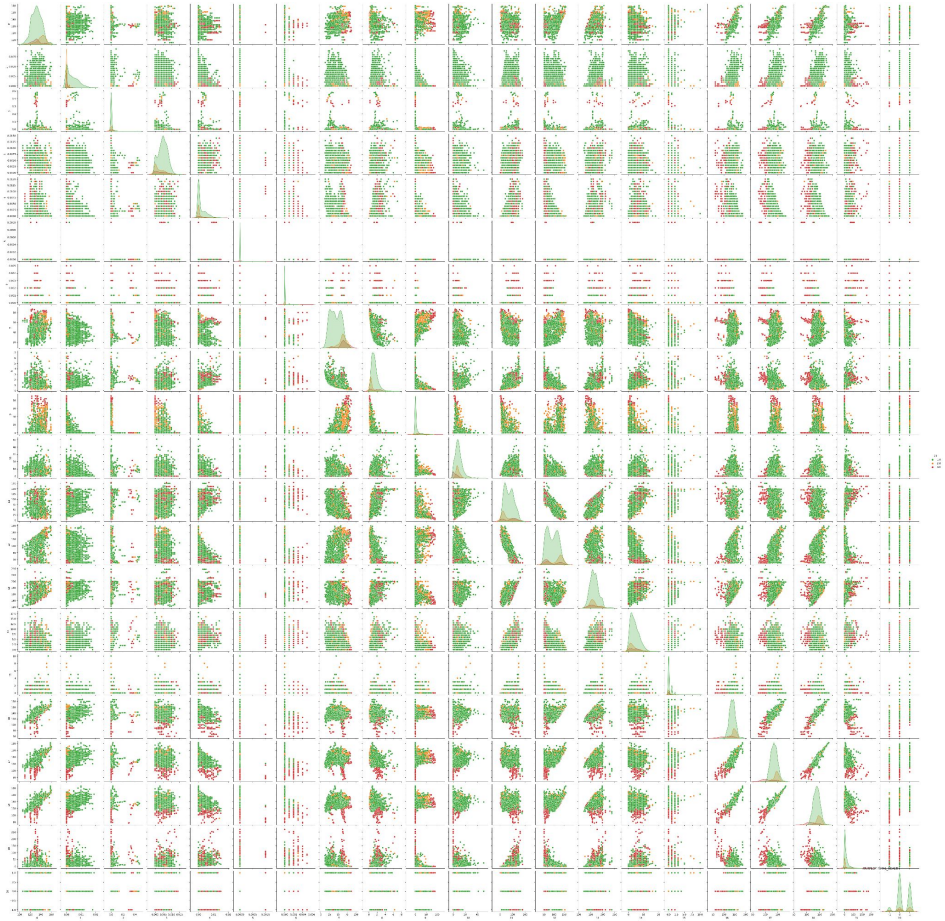
17 HISTOGRAM_MEAN

18 HISTOGRAM_MEDIAN

19 HISTOGRAM_VARIANCE

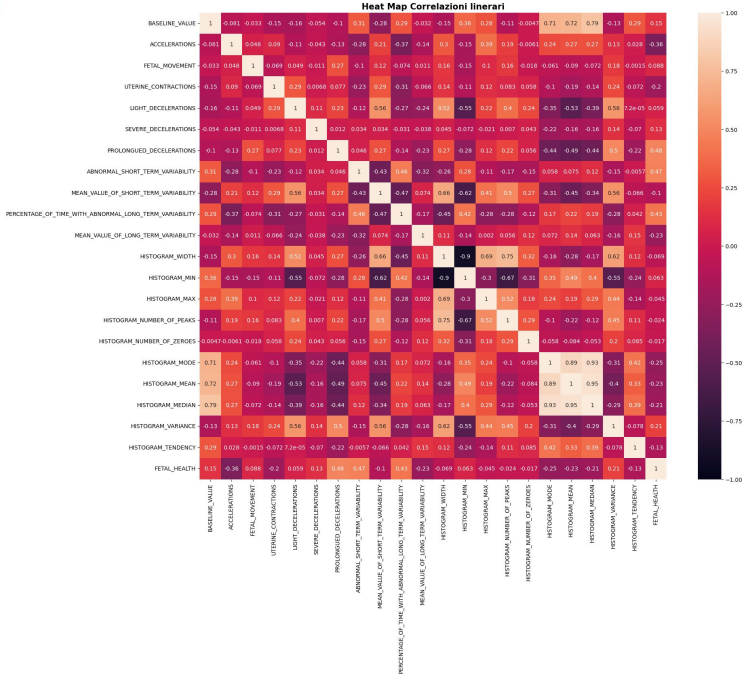
20 HISTOGRAM_TENDENCY

TARGET: 'FETAL_HEALTH'



EDA

Indici di correlazione tra **pair di features** incluso **target**



Analisi delle **features** con **quantità discrete**

unique value	UTERINE_CONTRACTIONS	LIGHT_DECELERATIONS	SEVERE_DECELERATIONS	PROLONGUED_DECELERATIONS	HISTOGRAM_NUMBER_OF_ZEROES
0.000	332.0	1231.0	2119.0	1948.0	1624.0
0.001	118.0	163.0	7.0	70.0	NaN
0.002	160.0	115.0	NaN	72.0	NaN
0.003	212.0	118.0	NaN	24.0	NaN
0.004	244.0	114.0	NaN	9.0	NaN
0.005	290.0	107.0	NaN	3.0	NaN
0.006	231.0	74.0	NaN	NaN	NaN
0.007	216.0	54.0	NaN	NaN	NaN
0.008	160.0	55.0	NaN	NaN	NaN
0.009	82.0	37.0	NaN	NaN	NaN
0.010	49.0	15.0	NaN	NaN	NaN
0.011	16.0	13.0	NaN	NaN	NaN
0.012	11.0	12.0	NaN	NaN	NaN
0.013	2.0	8.0	NaN	NaN	NaN
0.014	2.0	7.0	NaN	NaN	NaN
0.015	1.0	3.0	NaN	NaN	NaN
1.000	NaN	NaN	NaN	NaN	366.0
2.000	NaN	NaN	NaN	NaN	108.0
3.000	NaN	NaN	NaN	NaN	21.0
4.000	NaN	NaN	NaN	NaN	2.0
5.000	NaN	NaN	NaN	NaN	2.0
7.000	NaN	NaN	NaN	NaN	1.0
8.000	NaN	NaN	NaN	NaN	1.0
10.000	NaN	NaN	NaN	NaN	1.0

OSSERVAZIONI:
Eliminare la colonna 'SEVERE DECELERATIONS' sarebbe una scelta saggia in quanto contiene per lo più zeri e 7 valori identici, stesso discorso per **HISTOGRAM_NUMBER_OF_ZEROES** e **PROLONGUED_DECELERATIONS** ma prima di eliminarle verifichiamo se tutte le features sono importanti al fine di ottenere un modello accurato.

N.B. Le contrazioni dell'utero sono rilevanti per comprendere se un feto è normale ooppure patologico? Successivamente sarà verificato.

Modelli

- **Random forests**: come da traccia, è uno dei modelli ensemble omogenei + flessibili e non necessita di elaborazioni e ottimizzazioni delle features.
- **K-nearest neighbors (KNN)**: KNN potrebbe funzionare bene perché durante la EDA è possibile notare dei cluster pesati dal target.
- **XGBoost**: come da traccia.
- **Support vector machines (SVM)**: ricerca di un iperpiano tra le features per la separazione in classi, il dataset non è grande e quindi può avere delle buone performance.
- **N.P.** dovremo scalare le feature per far funzionare meglio SVM e KNN ma per ora verrà analizzato tutto senza modificare le features.

WORKFLOW

1. Divisione in **Train (80%)** e **Test (20%)** stratified
2. Grid Search Cross Validation sul Train + visualizzazione performance dei modelli ponderati da una lista di dizionari di iper-parametri.
3. Individuazione dei best iper-parametri per ogni modello.
4. Addestramento dei modelli ottimizzati e valutazione sul test.
5. Ripetere punti 3. e 4. se vengono modificate le features. (Verranno compiute 4 analisi complessive nel Notebook, quindi verranno fuori 4 tipi di train distinti)
6. Valutazione finale dei diversi modelli per ogni analisi compiuta nel punto 5.

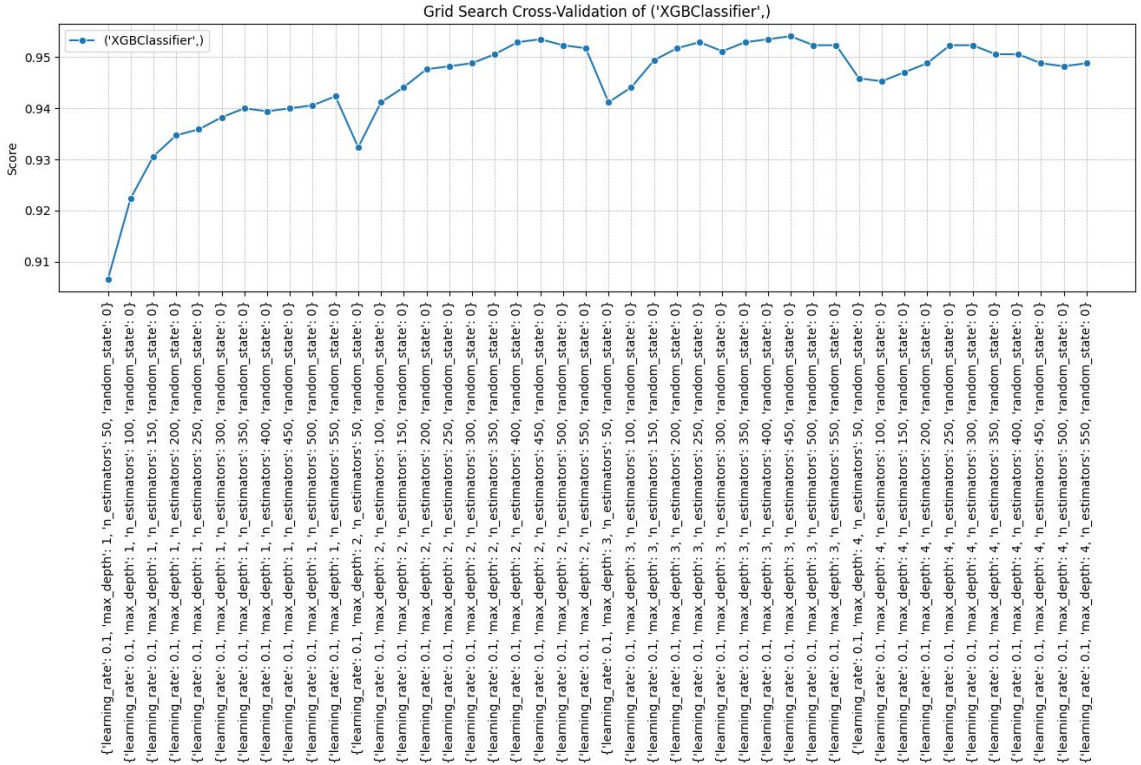
Funzione utile:

```
grid_search_cross_validation(models, params, X_train, y_train)
```

al fine di **ottimizzare il grid search** e il **cross validation** dei modelli è stata definita una **funzione che automatizza** e itera il seguente per ogni modello:

1. Esegua il GridSearchCV di scikit-learn sul train validato su 5 fold.
2. Mi salvi i risultati dello score per ciascun modello.
3. Mi restituisca un grafico per ogni modello valutato visualizzando lo score per ogni dizionario di iper parametri testati
4. Alla fine mi restituisce una lista con i risultati di ogni modello e un printaggio di una overview di tutti i miglior iper parametri trovati per ogni modello analizzato.

Esempio di Output di un grafico della funzione descritta:



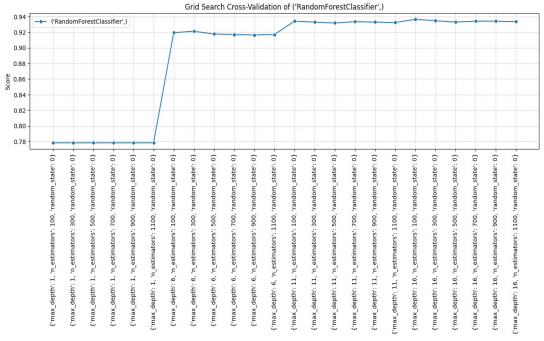
Esempio di Output alla conclusione dell'esecuzione della funzione:

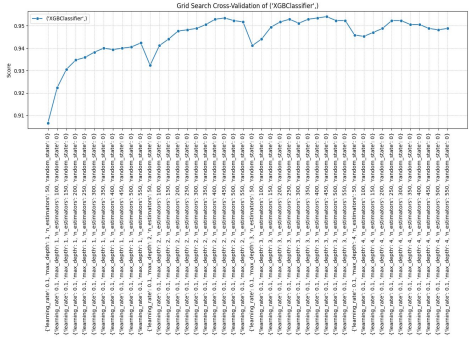
I migliori iper-parametri individuati del modello ('XGBClassifier',) sono: {'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 450, 'random_state': 0}, con un best score di: 0.95

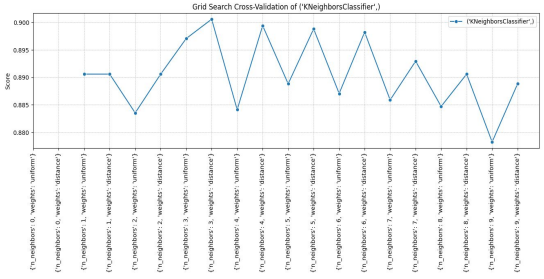
1° Model evaluation e feature importance

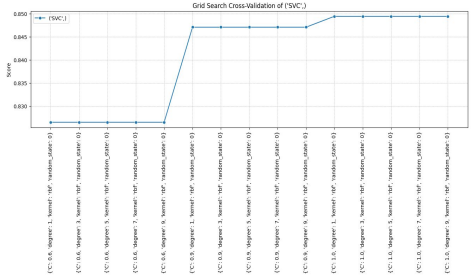
Il **primo train analizzato** (X_train_full nel notebook) contiene tutte le features senza modifiche.

Ogni grafico corrisponde a un modello analizzato, sono presenti gli score (in ordinata) della media della cross validation e (in ascissa) gli iperparametri corrispondenti applicati al modello.





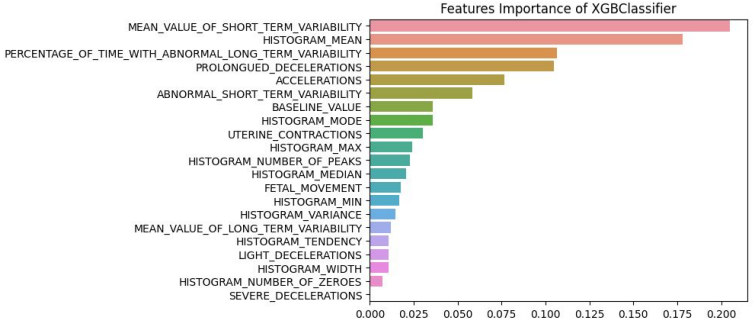




OSSERVAZIONI

- SVC: purtroppo non performa molto bene (85% di accuratezza), probabilmente le features sono da elaborare meglio per questo modello
- KNN: come era previsto, performa bene (90% di accuratezza) nonostante le features non elaborate.
- RandomForestClassifier performa meglio di KNN (93.6% di accuratezza) e qui non è necessaria una elaborazione delle features.
- XGBClassifier ancora meglio di RandomForestClassifier (95% di accuratezza) chissà come può performare dopo aver elaborato meglio le features.

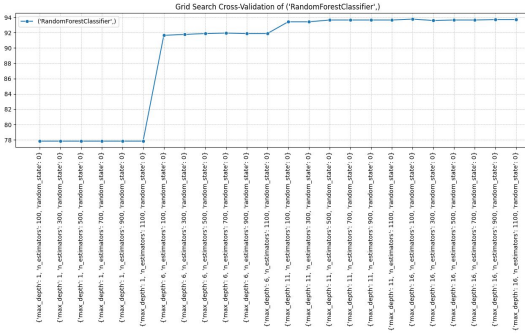
Feature Importance del XGBClassifier ottimizzato applicato al test.

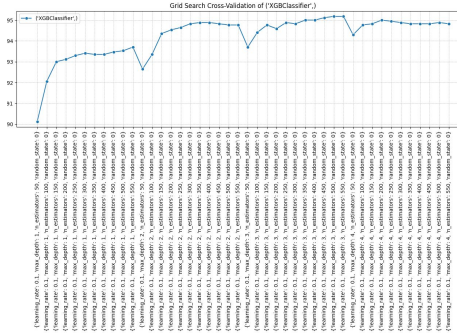


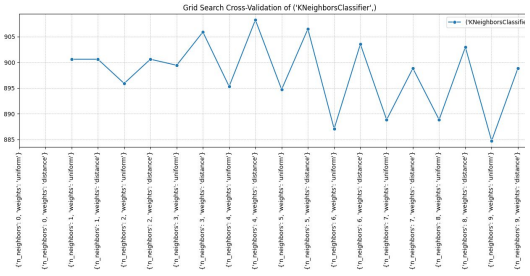
3° Model evaluation e feature importance

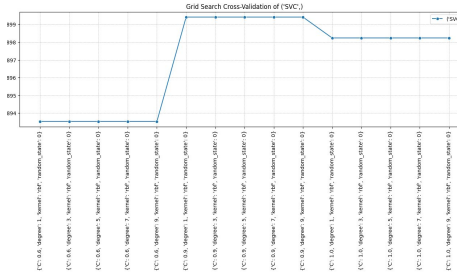
Il **terzo train analizzato** (X_train_full_v2 nel notebook, 2 perché contavo da 0) sono state rimosse le colonne 'SEVERE_DECELERATIONS' 'PROLONGUED_DECELERATIONS', 'HISTOGRAM_NUMBER_OF_ZEROES' e tutte le features sono state scalate da 0 a 1.

Ogni grafico corrisponde a un modello analizzato, sono presenti gli score (in ordinata) della media della cross validation e (in ascissa) gli iperparametri corrispondenti applicati al modello.





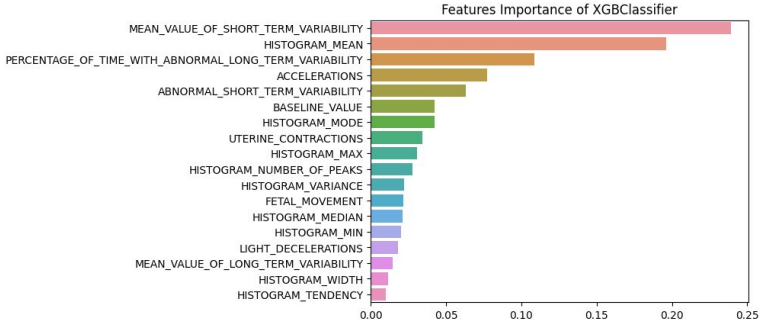




OSSERVAZIONI

- SVC: è migliorato molto rispetto alla prima analisi, avevamo un 85% di accuratezza e ora 90%.
- KNN: performa bene con 90.05% di accuratezza alla prima analisi ed è migliorato pochissimo con uno score del 90.82%
- RandomForestClassifier performa con 93.64% di accuratezza alla prima analisi e è migliorato leggermente al 93.76%.
- XGBClassifier era al 95.41% di accuratezza alla prima analisi, adesso performa con 95.17% ha un leggero peggioramento.

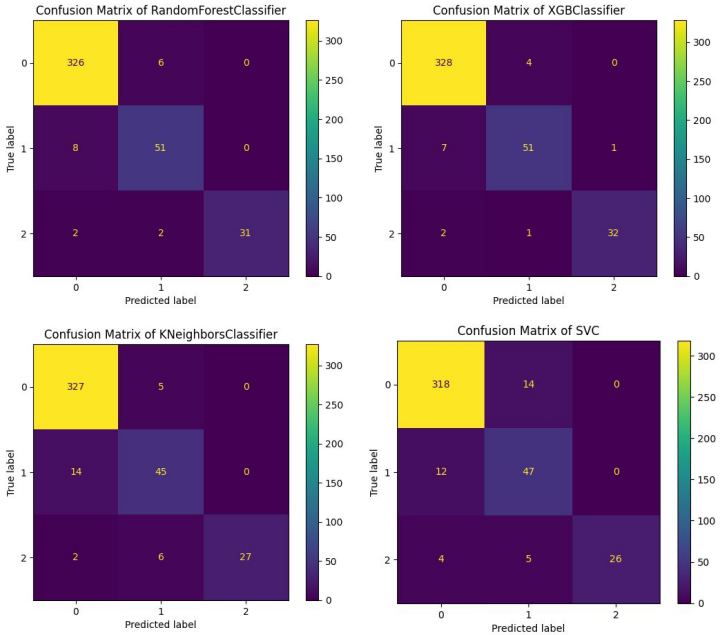
Feature Importance del XGBClassifier ottimizzato applicato al test.

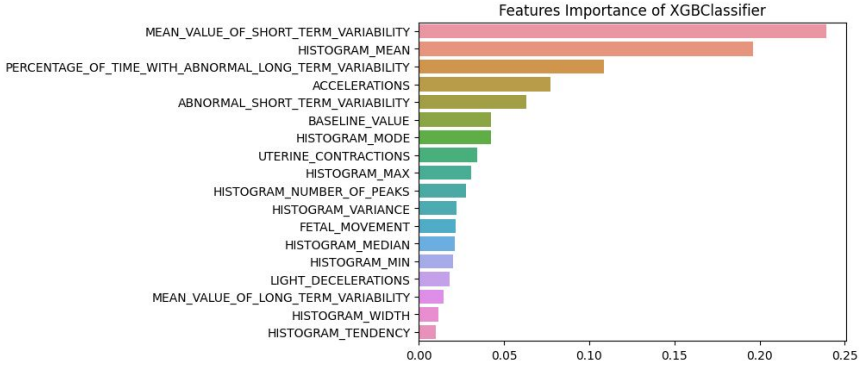


Conclusione

In questa slide è mostrata la valutazione sul test con confusion matrix di tutti i modelli ottimizzati con i migliori punteggi ottenuti sul train della 3° analisi (X_train_full_v2).

Il Test contiene 426 osservazioni con **332 feti sani**, **59 Sospetti** e **35 Patologici.**





OSSERVAZIONI FINALI:

- Sono state rimosse delle colonne poco utili:
  -'SEVERE_DECELERATIONS'
  -'PROLONGUED_DECELERATIONS'
  -'HISTOGRAM_NUMBER_OF_ZEROES'
- Le features sono state scalate tutte a valori compresi tra 0 e 1.
- Ne usciamo che **il modello che ha meglio predetto il test è XGBClassifier** {learning_rate = 0.1, max_depth = 3, n_estimators = 500}

XGBClassifier f1-score: **XGBClassifier accuracy: 0.96**

1. **Normal 0.98**
2. **Suspect 0.89**
3. **Pathological 0.94**